Datacomputer and SIP Operations: Semi-Annual Technical Report, January through June 1979

12

LEVEL II

Technical Report
CCA-79-22
July 31, 1979

Donald E. Eastlake III
Matthew Maltzman
Joanne Z. Sattley
Steven A. Zimmerman

Computer Corporation of America
575 Technology Square
Cambridge, Massachusetts 02139

12

6

Datacomputer and SIP Operations.

9

Semi-Annual Technical Report, Jan - Jun 79,

Technical Report

14 CCA-79-22

10

Donald E. Eastlake, III,
Matthew Maltzman,
Joanne Z. Sattley
Steven A. Zimmerman

12 58

11

31 Jul[redacted]79

15

This research was supported by the Advanced Research Projects ~~Agency of the Department of Defense~~ under Contract No. N00039-79-C-0203, ARPA Order [redacted] 3540. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Advanced Research Projects Agency or the U.S. Government.

387285

JB

Table of Contents

1. Summary

During the first half of 1979, Computer Corporation of America continued to provide its network-oriented database management utility, the Datacomputer, which is used by the seismic research community and by other users in diverse general networking applications. The Datacomputer's software system is designed to allow convenient and timely access to large on-line databases and to promote data sharing by multiple remote users communicating over a network.

The Datacomputer offers cost-effective large-volume storage capacity plus the ability to select and retrieve manageable subsets of data for efficient local processing. It is the only operational general purpose database system capable of manipulating data sets in excess of a trillion bits.

This very large storage capacity is made possible by the incorporation of an Ampex Terabit Memory System (TBM). The TBM at CCA is the only public installation of a data storage system based on videotape technology. The site is configured to hold up to 175 billion bits on line,

distributed over four TBM tape drives. Storage of data which may be referenced very seldom -- perhaps once in several years -- is managed on off-line tapes.

CCA operates and maintains the Datacomputer; as an attendant obligation, it responds quickly to the needs and problems of its users when they arise. The support functions include the operation of an on-line netmail help facility and 24-hour coverage of a telephone trouble line.

The intent is for the direct user of the Datacomputer to be a program running on a remote network host. CCA has implemented and continues to maintain a number of programs and subroutine packages which run on several ARPAnet hosts; these programs provide convenient access to the Datacomputer for remote users across the ARPAnet.

The interfaces range in scope from very low-level utility functions (which enable user programs to treat the Datacomputer as if it were a simple input/output device) to very sophisticated applications which exercise the Datacomputer's extensive indexing capabilities. Examples include the following: the DCLINK subroutine package for interfacing programs written in assembly language or BCPL to the Datacomputer; the DCPKG subroutine package for interfacing Fortran and COBOL programs; the RDC program for on-line input of Datacomputer instructions; the DFTP

program for simple, convenient transfer of local files to and from the Datacomputer; and the MARS Message Archiving and Retrieval Service. MARS users do not communicate directly with the Datacomputer, but rather, use ordinary network mail channels to transmit messages for filing and retrieving; "demon" programs operating on CCA-Tenex handle the actual storing and retrieving of messages on the Datacomputer as background processes.

Particular effort was directed in early 1979 toward designing and implementing a document storage and retrieval service along the lines laid out by MARS, but distinct in that the document system will perform direct file transfers over the network instead of relying on mail delivery. This document system, named DOCFILE, will be available later in the year for use by ARPA personnel and ARPA contractors in storing and retrieving official contract-related documents. The DOCFILE design specifications are included in this report as Appendix B.

The seismic application is the largest user of the Datacomputer. This is true not only in terms of the amount of data stored, but also in terms of its complexity and the bandwidth of transfer it requires. Some of the data involved is sent in real time to CCA, but not directly to the Datacomputer. The real-time data stream

is fielded by a small, reliable, dedicated processor, called the Seismic Input Processor, or SIP. This sturdy interface was designed and implemented by CCA; the SIP reformats received data for efficiency, and periodically forwards it in large batches to the Datacomputer.

Maintenance work on the Datacomputer itself, funded in part under Contract N00039-78-C-0443, engendered a new program release, Version 5/3, during this reporting period. The usual rigorous pre-release testing procedures were followed to ensure the upward compatibility of the modifications. Corresponding updates to the Datacomputer user documentation were also prepared and distributed.

The number of users and amount of data stored in the Datacomputer continue to grow. And by mid-1979, the half-trillion-bit level was surpassed.

2. Support of the User Community

A continuing and important task for CCA under this contract is to respond to queries about the Datacomputer from both the seismic and general user communities. Because the Datacomputer is a unique -- and evolving -- network resource, consultations, coordination and technical assistance are necessary if the best use is to be made of it. In the following subsections we give overviews of the seismic and general user communities and describe the significant developments in several CCA-provided Datacomputer user interface packages.

2.1 The Seismic User Community

The seismic users of the Datacomputer are primarily (but not exclusively) involved with research in the area of monitoring underground nuclear tests. Their main source of data is the worldwide network of seismic instruments, transmission lines, and data processors known as the VELA Network. Some of the components of the VELAnet are on the ARPAnet and use it as a data transmission system. Other

parts of the VELAnet use leased lines for real-time data or rely on the shipment of magnetic tape for non-real-time data.

The Datacomputer is the primary storage and retrieval resource in the VELAnet. The seismic data activity requires storage of very large amounts of on-line data including the following:

- seismic readings from arrays of instruments, some of which are processed in real time;
- seismic readings from individual Seismic Research Observatories (SROs) which are forwarded from the Albuquerque Seismological Laboratory;
- status and calibration information on instruments and on the VELAnet;
- derived seismic event summary information; and
- extracted signal waveforms corresponding to events.

The seismic instrument readings can be further divided into long period (one sample per second) and short period (ten or twenty samples per second) while the event summary information can be divided into preliminary and final versions. Since the Datacomputer is not designed to receive real-time data, a special dedicated miniprocessor,

the SIP, is used to buffer it as described in Section 4 below.

The status and event summary data are relatively compact and are of sufficient importance that they will be kept on line indefinitely. The seismic data readings, however, are so voluminous that they fill many reels of mass storage system video tape. Though most of these reels are, of necessity, kept off line, they can be mounted when needed on a day's notice.

In early 1979, CCA provided consultation and assistance to the seismic user community in utilizing the Datacomputer in the following noteworthy ways:

. In February, CCA representatives visited the Lincoln Laboratories Applied Seismology Group and the ARPA Nuclear Monitoring Research Office, and contacted the VELA Seismological Center to explore their future needs;

. In March, CCA consulted with the Seismic Data Analysis Center (SDAC) on their Datacomputer use, particularly regarding some network file transfer problems they were having;

. In April, the Albuquerque Seismological Laboratory (ASL) was assisted in accessing the Datacomputer for unusual maintenance operations on the files stored by ASL; and

. In May, CCA consulted with SDAC on ARPAnet bandwidth experiments which they were about to perform by sending additional real-time data from SDAC to CCA.

2.2 The General User Community

There are three broad catagories of non-seismic Datacomputer usage. These are:

- the on-line interactive interface programs;
- the program demons, and
- the special-purpose asynchronous-access application packages.

At present, on-line interactive use of the Datacomputer, the first category above, consists of the DFTP and RDC programs. DFTP connects in real time to the Datacomputer and enables its users to transmit local files into and from the Datacomputer. The RDC program is also a real-time iterface program; it is used mainly as a debugging tool for testing and refining Datalanguage instructions for later automated use by higher-level programs. RDC is used heavily by the CCA staff for pre-release reliability and regression testing of new Datacomputer versions.

There are a number of programs that automatically store data into the Datacomputer. Among these are the SURVEY system at MIT-DMS which stores information derived by polling the status of the ARPAnet server hosts, and the IMP-LOGGER system at BBN which stores information on events in the ARPAnet communications backbone.

The more sophisticated applications, MARS and DOCFILE, provide local interactive programs so that users may queue requests for archiving and retrieval services. The mail archiving activity, in particular, has stimulated mail system developers to include the concept of an archive button in their programs -- so that, by means of a single keystroke, users can designate that a copy of the message be sent to the archives. HERMES users have but to set up their templates appropriately in order to achieve a similar effect.

Although in the past it was possible to keep all non-seismic data in the Datacomputer on line, general usage has now grown to such an extent that an archival general purpose mass storage tape has had to be allocated and some rarely referenced files have been transferred to it. This archival tape is not always mounted.

2.3 User Interfaces

The Datacomputer is designed to be accessed only via ARPAnet connections; hence, communication with the Datacomputer requires a process running on some network host to handle the other end of the connections. CCA supplies and maintains a collection of subroutine packages and programs which provide easy access to the Datacomputer. A number of these are described below along with a summary of the pertinent developments in early 1979.

DCLINK and DCPKG provide Datacomputer-communication handling utility routines for inclusion in users' programs; RDC and DFTP, on the other hand, are complete programs which handle Datacomputer communications on one side, and offer simple interactive interface to a user terminal on the other.

2.3.1 DCLINK, a Subroutine Utility Package

DCLINK is a subroutine package that provides a convenient Datacomputer interface to programs on TENEX and TOPS-20 systems. It embodies many improvements in functionality, maintainability, and ease of use in comparison with its predecessor, DCSUBR. These are described in detail in CCA Technical Report CCA-79-11, Datacomputer and SIP Operations: 1978 Final Technical Report.

In recent developments, a shell has been implemented making DCLINK available by subroutine calls from the BCPL language. The previous release of the RDC program (based on the use of the DCSUBR package) has been superseded by an RDC based on DCLINK.

2.3.2 The DCPKG Subroutine Package

The DCPKG subroutine package provides for the use of the Datacomputer from Fortran and COBOL on the TENEX and TOPS-20 systems.

2.3.3 DFTP, the Datacomputer File Transfer Program

The function of DFTP is to offer a simple user interface, requiring very little knowledge of the Datacomputer system. DFTP provides for the storage and retrieval of whole local files between the Datacomputer and TENEX, ITS, Multics, TOPS-10, or TOPS-20 systems. In terms of the number of individuals making use of a facility, DFTP is the most popular Datacomputer application.

In the DFTP environment, a user's file collection is, in fact, generally stored in a single large file in the Datacomputer. The purpose is to generate files that more closely conform to the measurements expected by the Datacomputer's storage algorithms, which are optimized for large files. Users can group their files into hierarchically organized sub-directories; and the users are in turn grouped by their host or, in a few cases, by project or by host-group.

2.3.4 RDC, a Primitive Interactive Interface

As was discussed above, the user communicates with DFTP in terms of local filenames and command structures similar to those used for communicating with standard ARPAnet User-FTP programs. However, under some circumstances, usually involving an experimental or investigative activity, it is useful to provide a direct terminal connection to the Datacomputer with a minimal layer of intervening processing. RDC, now utilizing the *DCLINK* subroutine package, provides such an interface; it allows the direct input and execution of Datalanguage, and also has features to assist the user in establishing and managing separate data connections to the Datacomputer in conjunction with a terminal session.

In early 1979 RDC was modified to provide a larger Datalanguage type-in buffer, one the same size as the Datacomputer's input buffer.

2.3.5 MARS, a Message Archiving and Retrieval Service

MARS is made up of a package of programs which, working cooperatively, provide a unique way for ARPAnet correspondents to store messages on the Datacomputer, and to later retrieve selected subsets of the archived correspondence. Using the inverted-list indexing capabilities of the Datacomputer, each message is indexed on the field-names and keywords found in its header (separately by recognized fields), by date, and by words in the text-body; it may be retrieved by Boolean combinations of any of these.

The interactions between MARS users and the Datacomputer are accomplished by means of standard ARPAnet messages; the user h(im/er)self does not participate directly in the transmissions between MARS and the Datacomputer (except, of course, for providing the input and the motivation). Appendix A contains the essential MARS instructions in the form of an extractable User Card.

Standard mail-handling programs such as MSG and SNDMSG may be used for archiving individual pieces of mail by sending

copies to MARS-Filer at CCA. For archiving several messages -- or arbitrarily large collections of mail -- as a single operation, the newly developed MBQ program (a/k/a MARSBQ on some hosts) is available on Tenex and TOPS20 systems. MBQ constructs batch-mail input files and queues them for delivery to the MARS-Filer program operating on CCA-Tenex.

Although there are times when a filing backlog will accumulate -- typically around the first of each month when users batch-file their previous month's received mail, or when the Datacomputer's system is unavailable because of maintenance work and the like -- the usual case is for mail to be filed within an hour of its arrival at CCA-Tenex.

Retrievals are triggered by messages also, and it is possible to use any available message-composing program for this purpose. However, many systems offer the interactive RR program which is designed to assist users in preparing query-messages (also called "RRs", standing for Retrieval Requests). These messages, when delivered to the MARS-Retriever program operating on CCA-Tenex, are translated into Datalanguage requests which are then transmitted to the Datacomputer. The retrieved messages are mailed back to the requester and will appear as new mail in h(is/er) message file.

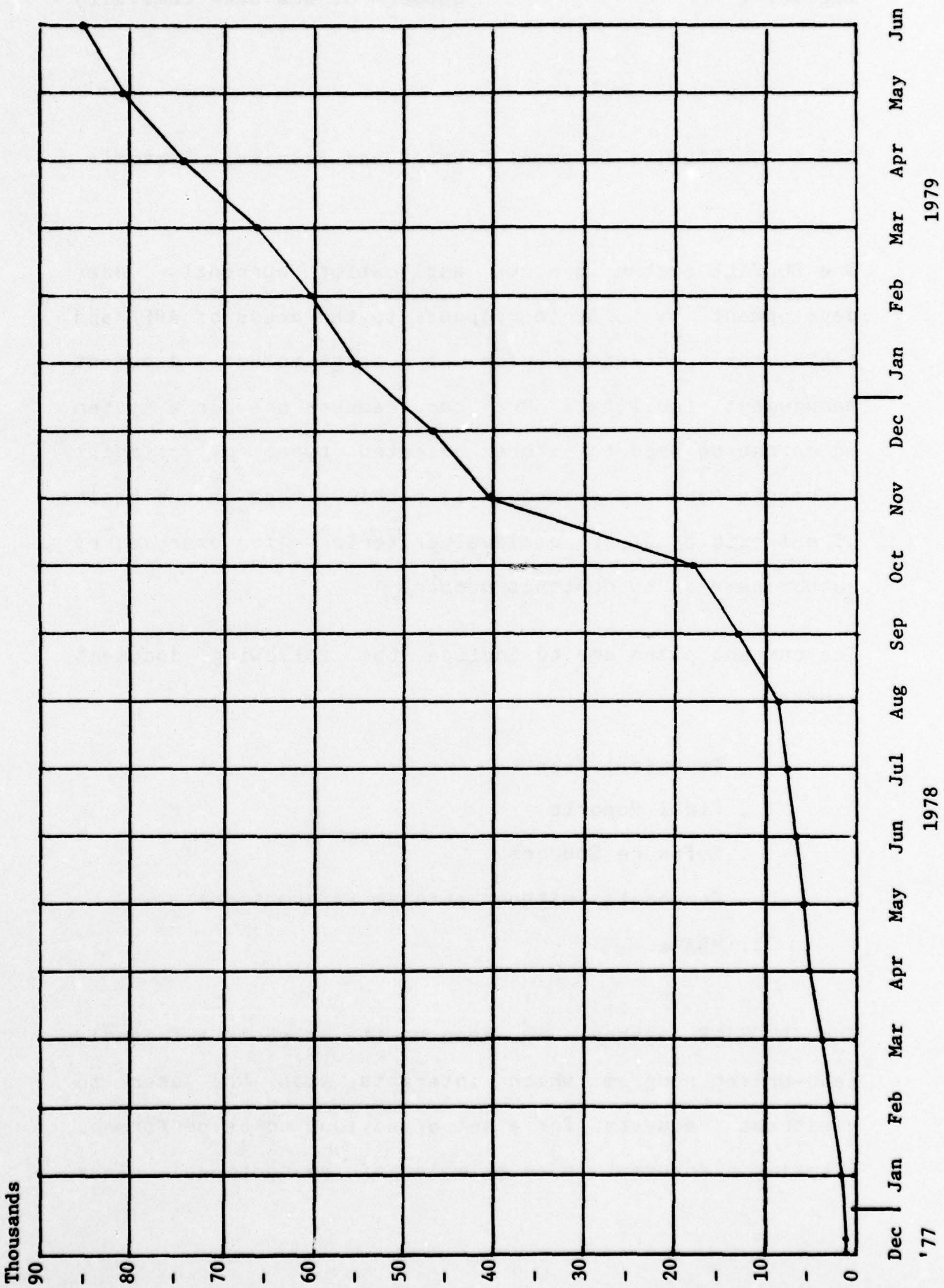
The MARS-Filer and MARS-Retriever programs operate as demons on CCA-Tenex. Each has a distinct network mailbox; each functions independently and asynchronously to perform its designated tasks.

The basic plan of MARS operations was originally distributed as Network Working Group RFC #744, NIC #42827. The first release of a functional package was described in 1978 as an informal report, MARS-Note #1. Although the program demons operate only on CCA-Tenex, the Service is available to any ARPAnet host that is able to send and receive mail.

In the first half of 1979, MARS was designated as the official public source for the ARPAnet MsgGroup correspondence. The synonym mailbox name "Public@CCA" is used for this purpose. The service has found acceptance, too, for the filing of private messages. These are messages whose distribution upon retrieval is restricted to the sender, the named recipients, and the archiver of the original message.

The MARS application has become steady tool, accepted and relied upon by the ARPAnet community. The growth of the message database is displayed below in Figure 2.1, a Cumulative Summary of Messages Filed through June 1979.

Mars: Cumulative Summary of Messages Filed — Figure 2.1



MARS: Cumulative Summary of Messages Filed

2.3.6 DOCFILE, a Document Storage and Retrieval System

The DOCFILE system is a new application currently under development by CCA in response to the needs of ARPA and contractor personnel for a large-volume document management facility. The requirements are for a system which can be used to store selected types of official documents and to subsequently retrieve them on the basis of a flexible set of retrieval criteria -- for example, by author name or by contract number.

The current plans are to include the following document types:

- Technical Reports
- Final Reports
- Software Sources
- Proposals (without pricing information)
- MRAOs

The DOCFILE system, as seen by its user, is a friendly menu-driven program which interacts with the user to construct requests for a set of actions to be performed. (Storing a document is an example of an action.) Each

action may be assigned priority (high, medium, low) by the user. The requested actions are appended to a work-list which may be examined by the user at will.

The DOCFILE system, in operation, is made up of an interactive user program and a background demon job. The demon program periodically scans the work-list, performs the requested actions in priority order, and posts the results back on the work-list. Acknowledgement of some actions will, in addition, be reported by net-message in the user's mail file.

The design specifications for the DOCFILE system are given in Appendix B.

3. Datacomputer Development

The Datacomputer is an operational system in which some evolutionary development is occurring. Version 5/2 of the Datacomputer was put into service in December 1978 and an update to the Version 5 User Manual describing the upward compatible changes it incorporated was distributed in early 1979. Version 5/3 is currently in service and incorporates an additional feature required and funded by the SDD-1 project. Version 5/4, which is currently being tested, is expected to be put into service shortly. The subsections below give further details of these developments.

For additional details on the structure of the Datacomputer and its development history, see Technical Report CCA-79-11, Datacomputer and SIP Operations: 1978 Final Technical Report.

3.1 Version 5/2 User Manual Update

In January 1979 a comprehensive update to the Version 5 Datacomputer User Manual, describing the developments in Version 5/2, was printed. Both the Version 5 manual and this update are 3-hole punched and designed so that the update can easily be incorporated in the manual. Developments described in the update include the rename option of the MODIFY command, the INHIBIT and EXHIBIT commands, the SUBSTRING function, and the REDESCRIBE command. Copies of the Datacomputer User Manual with all available updates can be obtained from CCA on request.

3.2 Version 5/3 Datacomputer

Version 5/3 of the Datacomputer was installed on 14 June 1979. It does not differ in any significant way from previous versions from the point of view of a regular user. Some internal improvements were included and a specially-accessed feature for limited reference to explicit file versions was added. This file-versions feature was requested and funded by the SDD-1 ACCAT project under a separate contract: N00039-78-C-0443.

3.3 Version 5/4 Developments

The following sections describe the Datacomputer developments incorporated in Version 5/4 which is currently undergoing pre-release testing.

3.3.1 Deleting Off-line Files

When deleting a file, it is necessary to free up the space the file used on its device. This is done by clearing the appropriate bits in the device's Volume Table Of Contents (VTOC), which normally resides on the device. The only device for which this is not true is the TBM, whose VTOC exists as a TENEX file. Before this year, the higher levels of the Datacomputer operating system did not know about this difference, and as a result, the Datacomputer always required all devices to be on line and mounted before any of their files could be deleted. This requirement was unnecessary in the case of the TBM, and needlessly prevented files on off-line TBM volumes from being deleted. Version 5/4 corrects this problem by allowing off-line deletes for TBM files only.

3.3.2 Permanent and Deletable Files

Previous to the Version 5/4 Datacomputer, any user could delete any number of files using the DELETE command, as long as s/he had the proper control privileges. Generally, the only time there was a potential problem here was when a whole class of files was being deleted with this command; it was possible to inadvertently delete an important file or node. In the Version 5/4 Datacomputer, the PERMANENT command has been introduced specifically to make files undeletable. Any user can make permanent any files for which he has control privileges. The command can be used on either single files or on whole subtrees, much as with the DELETE command. The DELETABLE command reverses the action of the PERMANENT command.

The PERMANENT command can also be used by the Datacomputer operator to make single nodes or whole subtrees permanent. The operator can use the PERMANENT command in either User or Operator mode. In User mode, the PERMANENT command works as described above, and any user with suitable control privileges can reverse its effect with the DELETABLE command. In Operator mode, nodes and files are

made permanent in such a way that only the operator can make them deletable again.

Finally, nodes can be made permanent in a third way by the Datacomputer itself. This would be done automatically, and would be used only for nodes and files which were essential for the operation of the Datacomputer, such as the directory file, which will be implemented in the next version of the Datacomputer. Once a node has been made permanent in this way, it cannot be made deletable again.

3.3.3 Unwritable Nodes

There are times when the Datacomputer operator may wish to freeze the data of a number of files belonging to a certain subtree of the Datacomputer directory. For this purpose, the Unwritable operator utility has been introduced. Once a node has been marked unwritable, none of the nodes in the subtree below it can be deleted. In this respect, it is like the operator version of the PERMANENT command. The Unwritable utility has the additional feature that files beneath this original node cannot have their data modified by write operations, although append operations are permitted. Subtrees can be made writable once again by use of this same utility.

3.3.4 The Utilities Subfork

Over the last few years, the Datacomputer program gradually grew in size until it reached the point where it completely filled its 256K-word core image. At this point, adding more features to the Datacomputer required that it be broken up into two separate processes, or forks, so that the resources of two complete memory spaces would be available. The decision was made to put all of the Datacomputer's utility routines in one fork and all the rest of the Datacomputer in the other. The main advantage of this approach is that the utility routines are used relatively infrequently, and as a result there is little overhead expended in switching from one fork to another. This change has been made in Version 5/4, and now there is sufficient room for both the current new routines and for future expansion.

4. The SIP

The SIP is a dedicated minicomputer communications system developed and operated by Computer Corporation of America. It interfaces real-time seismic information from the VELAnet to the Datacomputer.

Below we give a general description of the SIP and the changes that were developed for it in 1979. The general description given is similar to that in our Final Technical Report for 1978. Those familiar with the operation of the SIP in the VELAnet may safely skip it.

4.1 General Description

A primary function of the world wide VELA Seismological Network (VELAnet) is the collection of real-time seismic data from arrays of seismometers. This data is sent over leased lines and the ARPAnet to the Communications and Control Processor (CCP) at the Seismic Data Analysis Center (SDAC) in Alexandria, Virginia. From the CCP this data is immediately distributed to various processors and to the Datacomputer, via the SIP, for storage.

The components of the VELAnet that handle real-time seismic array data, except for the Datacomputer, are dedicated systems designed to receive data in real time. The Datacomputer, however, operates within a non-real-time operating system and serves the general ARPAnet community as well as the VELAnet. Furthermore, it is occasionally unavailable due to scheduled and unscheduled maintenance work. To isolate the Datacomputer from these real-time requirements, the SIP was implemented to receive real-time data from the CCP, to buffer and reformat this data on disk, and to periodically forward the data to the Datacomputer.

The SIP is implemented on a DEC PDP-11/40 computer. It has an ARPAnet interface, two RP04 disk drives for buffering data, an operator's terminal, and a status display screen. With the present bandwidth of data being sent over the network to the SIP and the present structuring of the SIP's disk storage, about two days of data can be held by the SIP.

Besides processing seismic data, the SIP software provides for operator communications between itself and the CCP. It also sends messages to the CCP for each chunk of data when the data has been properly filed in the Datacomputer.

4.2 SIP Development in 1979

The SIP is in an operation phase with very little development going on at this time. However, some minor changes were designed and implemented. The changes are expected to be installed later in the year.

Changes were made in the design of the SIP's directory structure, raising the maximum number of hours of data that can be stored on a disk pack from 32 to 128. The SIP in its present configuration of data sites will actually be able to store 49 hours of data on each pack, more than double its current 21 hours per pack. This increase is due partly to the directory design change and partly to the the cessation of the short period data and the data from the LASA site.

Other operationally motivated changes to the SIP that have been designed include the following: a change in the manner in which initial synchronization is achieved between the SIP's internal clock and the CCP's time source so as to speed synchronization; and a modification to the longer-term scheduling algorithms in the SIP so that it will tend to use the Datacomputer at night.

5. CCA-TENEX: Datacomputer Support

The Datacomputer runs as a user job on the CCA-TENEX ARPAnet host computer under the TENEX operating system. It is an unusually large and complex job composed of "subjobs" most of which serve remote users. Many modifications have had to be made in TENEX to accommodate the special requirements of the Datacomputer and the special hardware in use on the CCA Datacomputer system. The most prominent of these pieces of hardware is the Ampex TBM Mass Storage System.

Below we give a general description of the TBM system, a general description of the modifications we have had to make in the TENEX operating system, and a discussion of TBM and CCA-TENEX hardware and environmental problems thus far in 1979.

5.1 The TBM Mass Storage System

The CCA Datacomputer is equipped with the first public installation of the Ampex Terabit Memory (TBM) System. This device uses video tape technology to achieve a maximum on-line capacity of three trillion bits on up to 64 tapes. Maximum data transfer rate is 5.3 million bits per second.

The TBM at CCA is equipped with two dual tape transport modules so at most four tapes, or 175 billion bits (22 billion bytes) can be available on line. All equipment except for the four tape transports is non-redundant in the CCA configuration. This includes one Transport Driver (necessary for a tape to be in motion), one Data Channel (necessary to encode or decode digital information to and from the broadband analog signal on tape), one System Control Processor to coordinate and direct the other units, and one Channel Interface Unit that connects the TBM system to the Datacomputer's PDP-10 system. All of these units, which are non-redundant in the present CCA configuration, must operate properly for the TBM system to be usable by the Datacomputer.

5.2 TENEX Modifications

The modifications and additions that had to be made to CCA-TENEX to accommodate the Datacomputer include changes to both the operating system itself and to a number of separate utility programs that are not part of either the operating system or the Datacomputer proper. These modifications and additions include the following:

. improved efficiency in the network interface code for high volume file-transfer-like data streams sent and received by the Datacomputer;

. device code for using the Ampex TBM Mass Storage system and a set of CalComp 3330-equivalent disks which are used for "staging" -- intermediate storage between the PDP10 memory and the TBM;

. additional statistics-gathering code to aid optimization and analysis of operating system performance;

. a separate network server program augmented to provide status output on the Datacomputer;

. a utility program to run under TENEX for assisting in TBM maintenance;

. a utility program which runs all the time as a background job in CCA-TENEX, monitoring various system resources and alerting CCA personnel in case of problems; and

. additional utility programs for various purposes ancillary to the Datacomputer.

In January 1979, the CCA-TENEX system was modified to make full use of an additional RP-02 disk drive that had been purchased the previous year.

5.3 Hardware and Environmental Problems

The most severe environmental problem encountered during this reporting period was one of humidity control in the computer area. The TBM is particularly sensitive to low humidity. The humidifier in use was a sophisticated unit based on heating by electrodes immersed in the water. Unfortunately, the poor quality of water available led to extremely short electrode life, low reliability, and high maintenance cost. Replacement of the unit with one based on a simple, externally heated boiler has solved the problem.

Non-environmental problems included one failure of the CCA-TENEX disk controller, destroying the disk's directory structure, and some failures of the TBM controller.

The failure of the disk controller would have made the Datacomputer directory, which is stored as a TENEX file, inaccessible; however, a special utility program was implemented in less than a day which physically scanned all of the disk packs, one at a time, from the previous system, and recognized Datacomputer directory information. With a small amount of human assistance this utility reconstructed the entire Datacomputer directory. It continues to be the case that no normally stored user data has ever been lost by the Datacomputer.

None of the TBM controller failures caused any particular problem except for the unavailability, during repairs, of data which was resident only on tape. Data staged on the disk is generally still accessible, and data may be written up to the capacity of the staging disks without using the TBM.

A. Extractable MARS User Card

:: Archiving

Individual Private Messagese

. Include "MARS-Filer@CCA" on message distribution list (in CC:, FCC:, or BCC: field).

. Forward message to "MARS-Filer@CCA" [Additional keywords may appear in the Subject-field of the forwarding envelope.]

Individual Public Messages

. Include "Public@CCA" on message distribution list.

. Forward message to "Public@CCA"

Batches of Messages

. On TENEX systems, use the interactive MBQ.SAV program.

. On TOPS-20 systems, use the interactive MBQ.EXE program.

. On other systems, send the mail file as a single message to either "MARS-Filer@CCA" or to "Public@CCA" using the clue-word "batch" in the Subject-field.

:: Retrieving

. On TENEX systems, use the interactive RR.SAV program to prepare Retrieval Request messages and to mail them to "MARS-Retriever@CCA". The mail retrieved from the Datacomputer will be sent to the requester's mailbox.

. On TOPS-20 systems, use the interactive RR.EXE program.

. On other systems, send a message to "MARS-Retriever@CCA", specifying the retrieval criteria in the body of the message.

:: Sample Retrieval Criteria

```
SUBJECT:RFC#733 or RFC733      ; OR must be explicit

TEXT:MARS Project,goals        ; spaces & commas imply AND

DATE: 14 November 1977
SINCE: 1 Nov 77                ;same as AFTER:1 Nov 77
AFTER:1 Dec 1977
UNTIL: 15 January 1978         ; same as BEFORE: 15 January 1978

FROM:  JZS@CCA         ; host specification is optional
FROM:  FUSS,SYSTEM     ; comma implies OR (in FROM: field only)

TO: STEF@SRI-KA    ; host specification is optional
TO: SDD-0:,SDD-1   ; spaces and commas imply AND
```

:: Archiving

Individual Private Messages

.Include "MARS-Filer@CCA" on message distribution list (in CC:, FCC:, or BCC: field).

.Forward message to "MARS-Filer@CCA" [Additional keywords may appear in the Subject-field of the forwarding envelope.]

Individual Public Messages

.Include "Public@CCA" on message distribution list.

.Forward message to "Public@CCA"

Batches of Messages

.On TENEX systems, use the interactive MBQ.SAV program.

.On TOPS-20 systems, use the interactive MBQ.EXE program.

.On other systems, send the mail file as a single message to "MARS-Filer@CCA" for private mail or else to "Public@CCA". Use the clue-word "batch" in the Subject-field.

:: Retrieving

.On TENEX systems, use the interactive RR.SAV program to prepare Retrieval Request messages and to mail them to "MARS-Retriever@CCA". The mail retrieved from the Datacomputer will be sent to the requester's mailbox.

.On TOPS-20 systems, use the interactive RR.EXE program.

.On other systems, send a message to "MARS-Retriever@CCA", specifying the retrieval criteria in the body of the message.

:: Sample Retrieval Criteria

```
SUBJECT:RFC#733 or RFC733     ; OR must be explicit

TEXT:MARS Project,goals       ; spaces & commas imply AND

DATE: 14 November 1977
SINCE: 1 Nov 77               ;same as AFTER:1 Nov 77
AFTER:1 Dec 1977
UNTIL: 15 January 1978        ; same as BEFORE: 15 January 1978
BEFORE: AUG 7 76

FROM:  JZS@CCA          ; host specification is optional
FROM:  FUSS,SYSTEM      ; comma implies OR (in FROM: field only)

TO: STEF@SRI-KA   ; host specification is optional
TO: SDD-0:,SDD-1  ; spaces and commas imply AND
```

B. DOCFILE Design

This appendix describes the salient features of the DOCFILE DOCument FILing and retrieval system. This system was designed in response to ARPA's need for a high-capacity document management facility.

B.1 DOCFILE Requirements

Below are listed the five principal DOCFILE requirements and very brief comments regarding how the system is designed to meet them.

* The first requirement is the storage in digital form of the texts of large numbers of Technical Reports, proposals, and other ARPA contract and program-related documents. This storage will be initiated by ARPA and/or contractor personnel from several Arpanet hosts.

DOCFILE meets this need by providing an application package, available on each host, to give access to the Datacomputer storage facilities.

* The second requirement is a flexible retrieval facility for the stored documents. This includes a simple way to get a particular known document, and conditional retrieval based on various "header" fields associated with the document such as contract number, author, or title. It should be possible to retrieve either just headers or full documents.

DOCFILE meets these needs through the use of the application package mentioned above. The system will primarily use the indexing capabilities of the Datacomputer.

* The third requirement is that the user should be isolated from possible response delays or occasional unavailability in the Datacomputer.

DOCFILE accomplishes this by operating as a package of two distinct parts: a user program which primarily queues requests, and a background task which unqueues and executes them.

* The fourth requirement is for the protection of non-public documents. The desired method is to associate with users access authorization for documents related to particular contractors and / or ARPA programs.

DOCFILE will utilize the Datacomputer protection features to accept access only from certain host / socket-number combinations where authorized DOCFILE systems are running.

Access lists will be organized by contractor and host. Users will be identified by their login / connected directory name on each host supporting a DOCFILE package. The Tenex file system will be used for access list protection.

* The fifth requirement is the inclusion of a number of convenience features. These will assist the users by providing ways to examine the queue of requests and to cancel requests, and by providing name completion for some header fields. Such features will also tend to promote a uniformity of spelling and wording which will make conditional retrievals simpler and more effective. This is particularly important for contractor and program names on which the protection features are based.

DOCFILE will maintain files of 'seen' items for use by a name completion feature in the DOCFILE user program. Due to the importance of contractor and ARPA program names, lists of them will be held in the Datacomputer. These lists can be appended to and used to update similar lists at each host.

B.2 File and Data Element Design

This section describes the organization of the Datacomputer files which will be used by the DOCFILE system.

The document header information will be put into a single file, while the bulkier documents will be split among several files. This should decrease the necessity for tertiary-storage access since the most frequent accesses will be to the header file. A random access will be made to a document file only if a specific document from that file is wanted.

ARPA.DOCFILE.HEADERS

This is the one large header file to which all DOCFILE demons append new headers. It should be big enough to last for years. (IF we run out of space the program can be changed to reference a file group on retrieval. The HEADERS files would then need to be renamed and included in the group manually.) The Datalanguage used to create the headers file for the DOCFILE prototype is given below.

```
CREATE DOCFILE.PROTOTYPE.HEADERS FILE LIST(0,2000,20000),
                                                   IA=60000,ID=9
    HEADER STRUCT
        /* Fixed-length fields:  */
        ID BYTE,V=I
        PUB BYTE,B=1,I=D            /*PUBLIC*/
        DEL BYTE,B=1,I=D            /*DELETED*/
        SPARE1 BYTE,B=16            TYPE BYTE,B=18,I=D
        DOCID INT,I=D   /*DOCUMENT DOCFIL UNIQUE ID NUMBER*/
        CONTRACTOR INT,I=D
        ARPAPROG INT,I=D                    /*ARPA PROGRAM*/
        ARPAO STRING(5),S=ASCII,I=D         /*ARPA ORDER NUMBER*/
        ARPAL STRING(5),S=ASCII,I=D         /*ARPA LINE NUMBER*/
        CONTRACT STRUCT /* (integral number of 36-bit words) */
          ID STRING(20),S=ASCII,I=D        END /*CONTRACT*/
        DDATE INT,I=D   /*DOC DATE*/
        SDATE INT       /*STORE DATE & TIME*/
        SID STRUCT      /*ID OF STORER*/
          SITE  INT     /* <spare>b8+<host #>b17+<ttynum>b35 */
          LDIR  INT     /* login directory */
          CDIR  INT     /* connected directory */  END /*SID*/
        CDATE INT       /*CHANGE DATE & TIME*/
        CID STRUCT      /*ID OF LAST CHANGER*/
          SITE  INT     /* <spare>b8+<host #>b17+<ttynum>b35 */
          LDIR  INT     /* login directory */
          CDIR  INT     /* connected directory */  END /*CID*/
        QDATE INT       /*QUEUED D&T*/
        QID INT         /*QUEUE LOCATION*/
        DCNT INT        /*DOC LENGTH*/
        SYSTEM STRUCT /* (integral number of 36-bit words) */
          PROGSYS STRING(15),S=ASCII,I=D
          OPRSYS STRING(15),S=ASCII,I=D
          PROGSYSV STRING(6),S=ASCII
          OPRSYSV STRING(6),S=ASCII
          FILLER STRING(3),B=7,S=BINARY /* pad the last word */
          END
        SPARE2 INT,I=D              TITLESIZE INT
        ABSTRACTSIZE INT            WORDCOUNT INT
    /* Variable-length data:  */
        TITLE STRING(0,75,250),C=TITLESIZE,S=ASCII,I=D
        AUTHORS LIST(0,2,15),C=1
          AUTHOR STRUCT
            LASTNAME STRING(1,8,25),C=1,S=ASCII,I=I
            FULLNAME STRING(1,20,60),C=1,S=ASCII,I=I      END
        KEYABSTRACT STRING(0,250,3000),C=ABSTRACTSIZE,S=ASCII
        WORDS LIST(0,15,300),C=WORDCOUNT
          WORD STRING(1,7,29),C=1,S=ASCII,I=I
    END /*HEADERS*/;
```

ARPA.DOCFILE.DOCUMENTS.Hnnn.Dm

Document files are organized by host number "nnn". There will be a series of them for each host, distinguished by "m".

The Datalanguage used for creating the prototype system DOCUMENTS file is given below.

```
CREATE DOCFILE.PROTOTYPE.DOCUMENTS FILE LIST(0,20,100)
    DOC STRUCT
        DID BYTE,V=I
        SDATE INT          /*STORE DATE & TIME*/
        SID STRUCT         /*ID OF STORER*/
          SITE  INT        /* <spare>b8+<host #>b17+<ttynum>b35 */
          LDIR  INT        /* login directory */
          CDIR  INT        /* connected directory */
          END
        QDATE INT          /*QUEUED D&T*/
        QID INT            /*QUEUE LOCATION*/
        DCNT INT           /* 7-bit byte count */
        DOCUMENT STRING(1,50000,250000),B=7,S=BINARY,C=DCNT
    END /*DOCUMENTS*/;
```

ARPA.DOCFILE.PROGRAMS / ARPA.DOCFILE.CONTRACTORS

These are the global program/contractor name files. The Datalanguage used for creating this type of file for the DOCFILE prototype system is given below.

```
CREATE DOCFILE.PROTOTYPE.CONTRACTORS FILE LIST(0,100,1000)
   CONTRACTOR STRUCT
      NUMBER BYTE,V=I
      NAME STRING(120),I=D
      SDATE INT        /*STORE DATE & TIME*/
      SID STRUCT       /*ID OF STORER*/
        SITE  INT      /* <spare>b8+<host #>b17+<ttynum>b35 */
        LDIR  INT      /* login directory */
        CDIR  INT      /* connected directory */
        END
      CDATE INT        /*CHANGE DATE & TIME*/
      CID STRUCT       /*ID OF LAST CHANGER*/
        SITE  INT      /* <spare>b8+<host #>b17+<ttynum>b35 */
        LDIR  INT      /* login directory */
        CDIR  INT      /* connected directory */
        END
      QDATE INT        /*QUEUED D&T*/
      QID INT          /*QUEUE LOCATION*/
   END /*CONTRACTORS*/;
```

B.3 <DOCFILE> Files

The Queue

The DOCFILE Queue will be a series of TENEX files in the directory in which the DOCFILE demon program runs. A new version of the file will be created when all requests in the present queue are done and the file size exceeds 25 pages. The user program directly appends requests to the queue. The user will have read and append but not write access. The queue files are word-oriented files designed for easy access from BCPL.

Each queue file starts with the following fixed location words:

word	item
0	DOCFILE queue format number
1	Number of requests in queue that have been seen by the Demon
2	D&T of create
3	Version number
4	Host number
5	D&T of create of succeeding queue file

6-15 reserved

After the initial fixed words, the queue file has a series of requests. Each request consists of a request prologue followed by one or more subrequests. Words with zero value between requests are ignored. A request prologue has a fixed format as follows:

word	value
0	minus number of words in request
1	minus length of subrequest prologue (i.e., words to start of first subrequest)
2	minus number of subrequests in request
3	5 ASCII chars: first is L, M, or H for priority; remaining four are disposition of request as follows: NULL initial value from user program BFMT bad format, request ignored LOSE all of request failed WINS all of request successful CAND request cancelled MIXD subrequest dispositions mixed
4	D&T request appended to queue
5	D&T first subrequest started
6	D&T last subrequest done
7	Submittor's TTY line number

8	Submittor's login dir name, BCPL string
+	Submittor's connected dir name, BCPL string

To minimize problems resulting from system crashes, etc., requests are split up in such a way that no subrequest requires more than one Datacomputer interaction which modifies Datacomputer files.

Subrequests start with several mandatory fields. Some of these mandatory fields are for the demon to store information into the queue entry. Unless otherwise stated, these are initialized to zero. The mandatory fields at the start of a subrequest are as follows:

word	value
0	minus number of words in subrequest
1	word of five seven-bit bytes. The first is the retry count for the subrequest. The rest are disposition as for request above.
2	QID
3	D&T subrequest accepted
4	D&T subrequest completed
5	value returned by execution of subrequest, such as index for new contractor, DOCFILE document number for store text, etc.

6	spare
7	five ASCII characters designating subrequest type.
8+	additional fields depending on subrequest

Subrequest types recognized are as follows:

CANCL	Cancel queued request.
NCONT	New contractor name.
NPROG	New program name.
STORH	Store new header.
ADDHD	Add header for existing text.
STORT	Store new document text.
RETRE	Retrieve headers/documents.
UPDAT	Update header(s).

Arguments for each of these, following the mandatory fields, are given below.

CANCL: One word: Request number of queue entry to cancel.

NCONT: Contractor name as a BCPL string.

NPROG: Program name as a BCPL string.

STORH: 18 arguments corresponding to the fields in the Header file. They are (1) public bit, (2) deleted bit (always zero now), (3) document type, (4) document ID (may

be zero, see below), (5) contractor (may be zero, see below), (6) ARPA Program (may be zero, see below), (7) ARPA Order number, (8) ARPA line number, (9) contract number, (10) document date, (11) programming system, (12) programming system version, (13) operating system, (14) operating system version, (15) title, (16) authors, (17) abstract, and (18) key words. A zero means that the value should be filled in from a previous STROH, NCONT, or NPROG subrequest.

AUTHORS are separated by ";" and may contain space, comma, and period. A single author string is formatted as "<LASTNAME> <FULLNAME>;". WORDS are separated by spaces. The KEYABSTRACT field is last. The last four variable length fields are counted and a displacement is given for them.

ADDHD: Same as STORH except that the DOCID field is filled in initially.

STORT: A ASCIZ string which is the filename of the document text.

RETRE: (1) A one character flag field which is H if headers only are wanted and D if full documents are wanted. (2) A variable length string boolean expression which can be translated to Datalanguage for use in a WITH

clause. (3) A variable length string which is the file into which headers or documents are to be retrieved.

UPDAT: (1) A single character flag that is 1 if only one document should be modified and N if more than one can be modified. (2) A variable length boolean expression string suitable for use as a selecting WITH clause. (3) A variable length string translatable to Datalanguage to perform the update modification and suitable for use as the body of a FOR statement.

Program and Contractor List, Alias, and Access Files

In <DOCFILE> there will be two sets of three files. These sets are for distinguishing ARPA Contractors and ARPA Programs. One of the three files has the basic list of names, one per line, copied from the Datacomputer master name files. Another has a local list of aliases, one per line with each line starting with the program / contractor number (defined by position of name in basic name file). The third is the master access file. It has one line per program / contractor which contains the name of the local TENEX directory in which the local access file (if any) is given.

The directory specified in the master access file will have a file with a name which is a fixed function of the

program / contractor number and which lists all authorized users (indicated by directory name) and whether or not they are authorized to update things.

Name Completion

The name completion information will be four files in the <DOCFILE> directory. They are for (1) authors, (2) titles, (3) operating system names, and (4) programming system names. Each will be a simple text file with one entry per line. Any user can append to these files and read them.

Log File

The log file will be a text file appended to by various parts of the demon. When it gets too long (25 pages) a new version will be created.

B.4 Data Elements

DOCFILE Document ID Numbers

A DOCFILE Document ID number uniquely specifies the location of the text for a document in the set of documents files. It is a 35 bit quantity, usually printed in decimal with commas every three digits, as follows:

Bit 0	Unused, should be zero.
Bits 1-5	Check field, is twos complement sum of the rest of the word considered as six five bit fields.
Bits 6-14	Host number. Top bit zero and rest is old style number for now.
Bits 15-23	Document index within file.
Bits 24-35	File number for that host.

Document files are organized into a numbered sequence under a host node. The first documents stored from CCA are in ARPA.DOCFILE.DOCUMENTS.H31.D1.

Store and Change IDs and Times

The HEADERS, DOCUMENTS, PROGRAMS, and CONTRACTORS files have fields for the ID of the user originally storing a record and (except for DOCUMENTS) the most recent changer of the record. There are also date and time fields for these events. The date and time is simply the TENEX internal format. The format of the ID field is as follows:

Bits 0-8	Spare.
Bits 9-17	Host number. Bit 0 is 0 and the rest is old style host number for now.
Bits 18-35	TTY line number of user.
Next Word	Login directory number of user.
Last Word	Connected directory number of user.

Queue ID and Time

The HEADERS, DOCUMENTS, PROGRAMS, and CONTRACTORS files have fields for that queue location of the subrequest that originally created the entry. There is also a date and time field for the date and time the creating subrequest was queued. The date and time is TENEX internal format. The QID field is as follows:

Bits 0-8	Subrequest #
Bits 9-20	Request #
Bits 21-35	Queue file #

(note: host number is in SID)